

Developing a Decision Support System for Integrated Decision-Making in Purchasing and Scheduling under Lead Time Uncertainty

Benjamin Rolf
Otto von Guericke
University Magdeburg
benjamin.rolf@ovgu.de

Tobias Reggelin
Otto von Guericke
University Magdeburg
tobias.reggelin@ovgu.de

Sebastian Lang
Otto von Guericke
University Magdeburg
sebastian.lang@ovgu.de

Stefan Galka
Ostbayerische Technische
Hochschule Regensburg
stefan.galka@oth-regensburg.de

Abstract

Decision-making in supply chain management is complex because of the relations between planning tasks from different stages and planning levels. Uncertainties such as unpredictable supplier lead times and supply chain disruptions further complicate decision-making. Considering the case study of a company in printed circuit board assembly, a three-level concept is proposed that includes a decision support system. The global single-source supply network is characterized by highly variable lead times. Hence, the company maintains high inventory levels to prevent running out of stock. The decision support system considers the purchasing and scheduling decision problems in an integrated way. The prototypical implementation of the purchasing algorithm uses a genetic algorithm that recommends reorder days and order quantities using a simulation model. In addition, it evaluates the risks of the recommended solution by calculating the probability of stockouts for each order cycle.

1. Introduction

Supply chain planning involves complex decision situations with several decision alternatives that are often difficult to define, and those impacts are partly unknown [1]. Planning problems in supply chain planning are not isolated but rather highly interconnected [2]. Therefore, it is often hard for decision-makers to assess the situation and choose the correct actions. Especially in small and medium-sized enterprises the availability of data is rather low. Although they can have strong impacts on the supply chain performance, decision-makers often rely on their domain knowledge when making decisions. In addition, uncertainties impact the decision-making in real supply networks. The company in the considered case study has a global single-source supply network for each material and experiences highly varying lead times. The variable lead times result in high inventory

levels in order to secure the raw material availability for scheduling. Therefore, the company needs a decision support system (DSS) that recommends actions regarding purchasing decision-making and should also be aware of risks and transparency.

In the following sections, we first review relevant literature on multi-stage planning problems and uncertainties on the procurement stage. Then, we define the purchasing and scheduling problems as well as their interface in section 3. Section 4 presents a conceptual modular DSS that is based on a cyber-physical system (CPS) architecture. The purchasing algorithm which optimizes the purchasing problem is described and implemented in sections 5 and 6. Last, we draw a conclusion and state future research directions.

2. Literature Review

Supply chain planning under uncertainty has received increasing attention in literature in the past years. Chopra and Sodhi [3] investigated different categories of supply chain risks, their drivers and mitigation strategies. Risks occur on all stages and planning levels of supply chains, and they often depend on each other. Hence, effective planning and risk mitigation require to consider the procurement, production and distribution stages of the supply chain in an integrated way.

Several authors formulated and solved integrated production-distribution planning problems. Ben Abid et al. [4] considered the integrated optimization of production and distribution with uncertain production capacities and customer demands. The production network consists of four stages and represents the system of a company in Tunisia. Karimi et al. [5] formulate an optimization problem considering the production and distribution stages with manufacturers, distributors and retailers. They also include the routing problem and consider transportation cost discounts. Khalili and Farib [6] focus on resilience and uncertainty in production and distribution and also consider strategic

decisions to decrease the impact of supply chain disruptions.

Some authors also considered the three-stage problem including procurement, production and distribution. Wang [7] used particle swarm optimization to solve a deterministic three-stage supply chain optimization problem with randomly created test instances. Park et al. [8] apply a genetic algorithm to solve a deterministic three-stage problem with known demands and lead times. Torabi and Hassini [9] present an optimization model for procurement, production and distribution planning considering the case study of an automobile company. Kanyalkar and Adil [10] consider the case study of a consumer goods company.

However, the presented papers address deterministic problems and therefore neglect uncertainties in the procurement stage. Su and Liu study the supplier selection under uncertainty in a dual-sourcing problem with a local and a global supplier. Ramasesh [11] emphasizes that uncertain lead times result in higher inventory costs and proposes a dual-sourcing technique for supply networks with uncertain lead times. Ammar et al. [12] provide a literature review on supply planning under lead time and demand uncertainty. Saleh Sadghiani et al. [13] consider the strategic supply chain network design with the goal of minimizing the impact of supply chain disruptions on the procurement stage.

Optimization Models that consider the integrated optimization of procurement and production with uncertain lead times are rare. Fallah-Jamshidi et al. [14] and Hnaïen et al. [15] investigate a two-stage assembly system with lead times that are sampled from probability distributions. They implemented a genetic algorithm that aims to minimize inventory costs and to maximize the customer service level. Especially the influence of purchasing decisions in global supply chains with a single source on machine scheduling requires further research.

3. Problem Definition

The case study of this paper considers the supply network and the production system of a medium-sized company that manufactures printed circuit boards in a single production site in Germany (see Figure 1). Both, suppliers and customers, are big players with strong market positions which leads to a difficult position in the value chain for the company resulting in delayed deliveries on the supplier side as well as impending contractual penalties on the customer side. Decision-making in this context requires a scope that is beyond a single domain but rather considers multiple planning problems. Contrary to similar case studies

in literature, this problem includes both, a multi-stage scope and stochastic demands and especially lead times. Decisions in both domains, purchasing and scheduling, influence each other and require integrated planning. However, integrated planning does not necessarily require merging both domains into a single decision problem but rather synchronizing the inputs and outputs. Hence, we first formulate the static decision problems of both domains and then present a DSS framework in section 4 that adapts them to the dynamic real-world environment.

3.1. Purchasing Problem

Considering the purchasing domain, the decision-makers have to determine a set of orders for each material m . An order consists of exactly one reorder date R_m which determines the day on which the order is placed and exactly one order quantity Q_m . Currently, an enterprise resource planning system provides the information for purchasing planning. The system uses fixed lead times that the decision-makers have to update manually. However, representing the orders with fixed lead times is inadequate in this case study because the lead times are highly varying according to the empirical knowledge of the decision-makers. In addition, the demands vary from day to day. Therefore, we define the following characteristics for each material:

- The lead time LT describes the required time from order until arrival in the warehouse. The lead times are subject to a stochastic distribution.
- The demand DE_d on day d describes the daily demand on a specific day in the planning horizon. The daily demands are subject to a stochastic distribution.

With an arbitrary set of orders and the material characteristics it is possible to calculate the daily inventory levels of the materials. The inventory level IL_d on day d defines the amount of materials in the warehouse on the specific day. It is assumed that inventory changes are not staggered within a day. Hence, it is sufficient to capture the inventory level once a day. Furthermore, negative inventory levels are permitted in order to represent unmet demand. The array of daily inventory levels then yields the inventory development in the planning horizon and serves as the basis for calculating objective values.

The purchasing strategy of the company focuses on the major objective of maximizing the material availability and the minor objective of minimizing the inventory

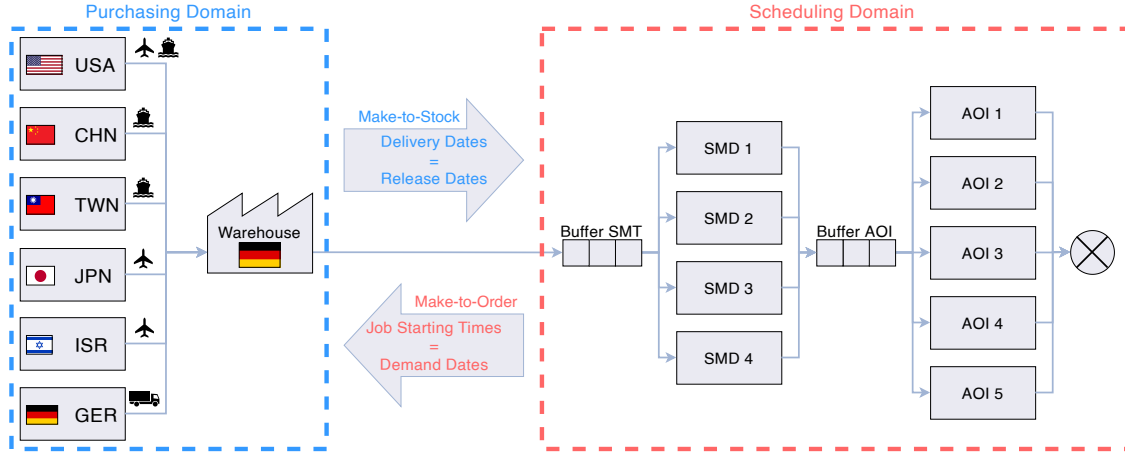


Figure 1. Delimitation of the purchasing and scheduling domains considering the supply network and production system of the company.

costs. Both objectives are in general highly contrary because low inventory levels result in low material availability considering stochastic systems and vice versa. Optimizing both objectives requires a weighted multi-criteria objective function. Therefore, we propose an objective function that uses three cost rates to rate the inventory levels of each day in the planning horizon:

- The cost rate c_{inv} applies to inventory levels that are above the safety stock of the target material.
- The cost rate c_{saf} applies to inventory levels that are below the safety stock target but still positive.
- The cost rate c_{tar} applies to negative inventory levels, respectively unmet demand.

Multiplying the respective cost rate with the inventory level of the specific day yields the daily inventory costs IC_d for day d :

$$IC_d = \begin{cases} c_{inv} * IL_d & IL_d \geq SS \\ c_{saf} * IL_d & SS > IL_d \geq 0 \\ c_{tar} * IL_d & < 0 \end{cases}$$

Accumulating the daily inventory costs for the entire planning horizon ($t = [t_s, t_e]$) results in the objective function for the minimization problem. The objective function yields an objective function value for each solution s :

$$\min f(s) = \sum_{d=t_s}^{t_e} IC_d$$

3.2. Scheduling Problem

Considering the scheduling domain, the decision-makers have to determine a machine schedule. The production system consists of nine machines that are arranged on two consecutive stage. The first stage consists of four identical parallel surface mounting devices (SMD) and the second stage consists of five identical parallel automatic optical inspection devices (AOI). Each job has to pass through both stages in the same sequence. Hence, the production system is a two-stage hybrid flow shop. Each job j has a release date r_j , a due date d_j , processing times p_{1j} and p_{2j} and a product family f_j .

The company clustered similar variants in product families in order to simplify the setup process on the SMD stage. As each setup cart configuration is able to produce all variants of a product family, the changeover process is only required when switching the processed product family. Therefore, the duration of a setup depends on the fact whether the next job is from a different product family than the previous job. The setup times are therefore family-dependent:

$$s_{j,k} = \begin{cases} 20(\min) & f_j = f_k \\ 65(\min) & f_j \neq f_k \end{cases}$$

The setup times on the AOI stage are sequence-independent:

$$u_{j,k} = 25(\min) \forall j, k$$

When evaluating the quality of a schedule, the company attempts to achieve two objectives, namely minimizing the total tardiness and minimizing the

makespan. A machine schedule should optimally not include tardy jobs at all because the contractual penalties for late delivery are high. The makespan in a hybrid flow shop with family-dependent setup times depends on the total processing times, minor setup times and major setup times. Minimizing the makespan implicitly comprises two subobjectives which are achieving an even utilization on all machines and minimizing the sum of setup times [16]. The current family production strategy solely focuses on the minimization of setup times. Therefore, it neglects balancing the utilization leading to suboptimal machine schedules. We define the sum of makespan and total tardiness as the objective function for the minimization problem:

$$\min g(s) = C_{max} + T_s$$

3.3. Interface

Purchasing and scheduling problems have been studied extensively in the past years, but the interface between both has not received a lot of attention yet. However, broadening the scope can possibly improve the coordination of actions in both domains in order to support the decision-making.

The production principle of a company, either make-to-stock (MTS) or make-to-order (MTO), has a major impact on the interface because it determines the temporal sequence of planning. If a company produces according to the MTS principle, the planning process depends on the supplier side. Whenever the company receives incoming goods, it can trigger production which in turn triggers distribution and sales after finishing. The release date of a job then depends on the inventory levels of the materials, and it is equal to the latest availability of all required materials. If a company applies the MTO principle, each job belongs to a specific customer order with a due date, and the planning process depends on the customer side. The sales stage first provides job information to the production stage that generates a machine schedule and then the procurement stage has to ensure the in-time supply of materials. In this case, the job starting times determine the demand dates for purchasing which are the latest possible delivery dates. If a company applies the MTS principle, it determines the purchasing plan first and afterwards schedules the jobs. The delivery dates determine the release dates of the jobs.

Synchronizing release dates and delivery dates in MTS, respectively job starting times and demand dates in MTO, has a significant impact on inventory management. The closer the two dates are together, the lower is the inventory level in the warehouse which results in lower inventory costs. However,

in a real-world environment with variable lead times and demands like in this case study, it also results in higher risks of running out of stock because low inventory levels cannot buffer uncertainties in the supply chain. This trade-off is the central factor for successful operations considering the interface of the purchasing and scheduling domains. In the next section we propose a concept that is tailored to the MTO production of the considered company.

4. Conceptual Decision Support System

The company requires a tool that supports the decision-making in the purchasing and scheduling domains. The ever-changing data basis in these dynamic domains results in quickly outdated information. Therefore, static optimization models are not sufficient for deployment in real manufacturing environments. As the considered production system still operates manually, we propose a comprehensive concept that does not only include the DSS but also a potential CPS architecture. The concept is based on a three-level architecture with the physical level, the information level and the analytics level (see Figure 2). Focusing on the analytics level which contains the core DSS, the roles of each level are described in the following subsections.

4.1. Physical Level

The physical level includes all machines, devices and structures that are concerned with the material flow such as the supply network, the warehouse and the production system. It especially requires considerations on hardware that is capable of establishing a connection between the physical production system and the digital shadow. A digital shadow implements an automated one-way data flow between physical system and digital system [17]. Therefore, the concept proposes potential locations for sensors that keep track of state changes in the CPS in order to associate physical objects with their digital representations. The sensors have to acquire data on the job that each machine currently processes and its remaining processing time, the current setup of each setup cart and its position and the state changes of machines and their tools. Identification and localisation require to assign each job a unique identifier, for example by means of a barcode. However, jobs are not physical objects, therefore they can only be represented by the setup cart that is currently setup with the specific raw material. Equipping the machines and buffers of the SMD and AOI stages with vision-based sensors could then enable the identification of a job when the corresponding setup cart enters the detection zone.

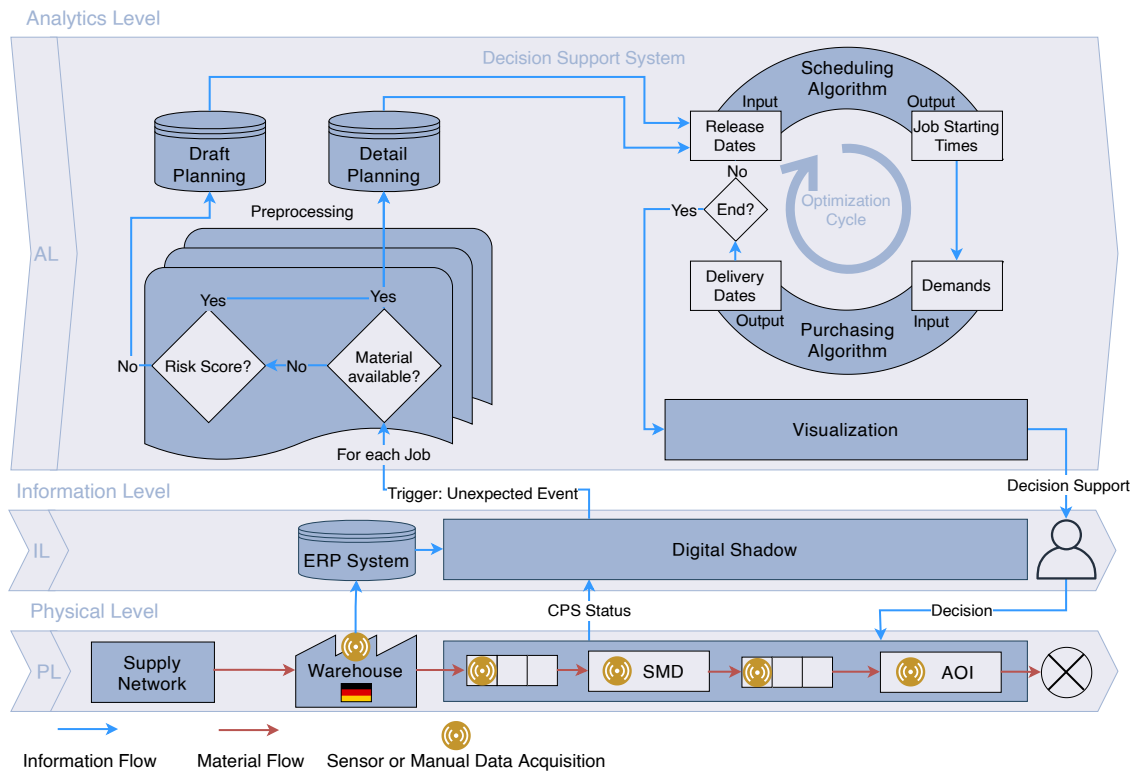


Figure 2. Concept for a modular decision support system based on a cyber-physical system architecture in a make-to-order environment.

4.2. Information Level

The information level handles the information flow and includes all software tools that store and provide data such as the digital shadow and the ERP system. It provides a structured data basis with standardized interfaces for external tools such as the DSS. In addition, the concept depicts the human decision-maker on the information level because he significantly contributes to the information basis with his domain knowledge, and he has to make the final decision. Considering the DSS that should support but not automate decision-making, the concept explicitly includes a digital shadow whose data flow is automated in the direction of data acquisition but not vice versa. The power to utilize the outputs of the DSS and implement decisions remains completely at the decision-maker.

4.3. Analytics Level

The analytics level includes all data analytics applications that process data from the information level. It contains the DSS which processes data from multiple sources and gives feedback to the decision-maker in form of recommendations. The DSS

has to fulfill the following requirements of the company:

1. It should output concrete recommendations on how to proceed regarding purchasing and scheduling originating from the current status of the supply network and the production system. These recommendations should include concrete values for the decision variables of the purchasing and scheduling problems.
2. Due to the high importance of those decisions, it should make the recommendations as transparent as possible to make the decision-maker aware of risks and create a basis of trust.

The DSS consists of the data preprocessing module, the optimization cycle and the visualization module. The goal of the data preprocessing module is to split all currently known jobs in a detail planning set and a draft planning set. The detail planning set should contain jobs for which all materials are available in the warehouse or for which the risk of late delivery is low, whereas the draft planning set should contain the remaining jobs. To split the jobs, the algorithm checks the material availability and the risk score of each job. The risk score has to be determined for each material

and could for example depend on the supplier reliability or the expected lead time. The optimization cycle then takes both sets as separate inputs.

The optimization cycle is in charge of creating solutions for the decision problems and synchronizing them. Its design is modular because it should include arbitrary algorithms for both domains as long as they can deal with the stated inputs and outputs. As the company uses the MTO principle, the optimization cycle first starts the scheduling optimization and generates a schedule. The scheduling algorithm then uses the jobs in the detail planning set to create a concrete schedule, whereas the other jobs are only considered to block sufficient capacity at the end of the schedule. After the scheduling is done, the purchasing optimization begins using the starting times of the jobs in the schedule as demand dates. The purchasing optimization in turn generates a purchasing plan with expected delivery dates that either suit the job starting times in the schedule or not. The termination criterion then checks whether a solution is suitable or not. If the solution is suitable, the planning cycle terminates, if not it is repeated until it does. After finishing the optimization, the visualization module visualizes the solution, for example in a graphical user interface that is presented to the decision-maker.

As the mathematical problem formulations do not directly represent the reality, the DSS architecture has to face two difficulties: The mathematical problem definitions consider static problems, but in reality the data basis for these problems is continuously changing. Supply chain management in general is a rolling process and the company exists for several years. Hence, the problems are currently not in an idle state but rather in a steady state with existing schedules and pending orders that are continuously updated. However, the static problems are only suitable for the real-world environment if the optimization is repeated whenever an unexpected event occurs. Fleischmann et al. [2] refer to this as event-driven planning. In this case, every unexpected event causes the full computational effort of optimizing both problems again. Therefore, this procedure is only suitable if unexpected events only occur occasionally, and fast optimization algorithms are available. Uncertain events in this case study are either late deliveries or newly released jobs. Both events trigger the optimization cycle but happen no more than a few times a day.

The solutions of the purchasing and scheduling problems are highly dependent and require an integrated view. But instead of integrating the optimization of both problems, the optimization cycle aims to rather synchronize it. When synchronizing the problems, the optimization cycle can either start by generating

a schedule and then determining a purchasing plan depending on the schedule or vice versa. Both ways are viable in general, but due to the MTO production of the company, the operational planning traditionally starts from the customer demand side. The customer demands and due dates are fixed first. Next, the scheduling is done based on the fixed demands, and in the end the procurement stage determines a purchasing plan to meet the fixed schedule.

5. Purchasing Algorithm

Multiple static scheduling algorithms have been proposed in the past years for this production system that the optimization cycle can possibly use. These include reinforcement learning [18], problem-specific heuristics [19] and a dispatching-rule-based genetic algorithm [20]. As the purchasing domain was not considered in the past, we propose a novel purchasing algorithm that can be integrated in the optimization cycle.

The algorithm has to fulfil two requirements: It has to be fast to fit in the optimization cycle, and it has to be capable of dealing with small and heterogeneous datasets. The performance of algorithms is highly dependent on the specific application domain and therefore the selection of an algorithm is an important part of the concept. In this case, finding an algorithm that suits the requirements was a trial-and-error process. Initially we planned to implement a data-driven algorithm such as a neural network or support vector machine but they require more data than available in this environment. Hence, they turned out to be unsuitable for this problem. Metaheuristics seemed to be more promising because they do not require historical data but rather run a specific optimization for each new problem instance. This is an advantage regarding the necessary data but a disadvantage considering the computation time. Out of the metaheuristics domain, evolutionary algorithms provide the best capabilities to adapt them to specific problems, therefore the purchasing algorithm is based on a genetic algorithm. A discrete-event simulation model and a risk evaluation module complete the architecture (see Figure 3). The purchasing algorithm optimizes only one material at a time and therefore has to be run once for each material.

5.1. Lead Time Simulation Model

The discrete-event lead time simulation model consists of a probability density function that generates random lead times for each order, and it calculates the expected inventory development from the orders and demands. We used a kernel density estimation to create the probability density function. The simulation

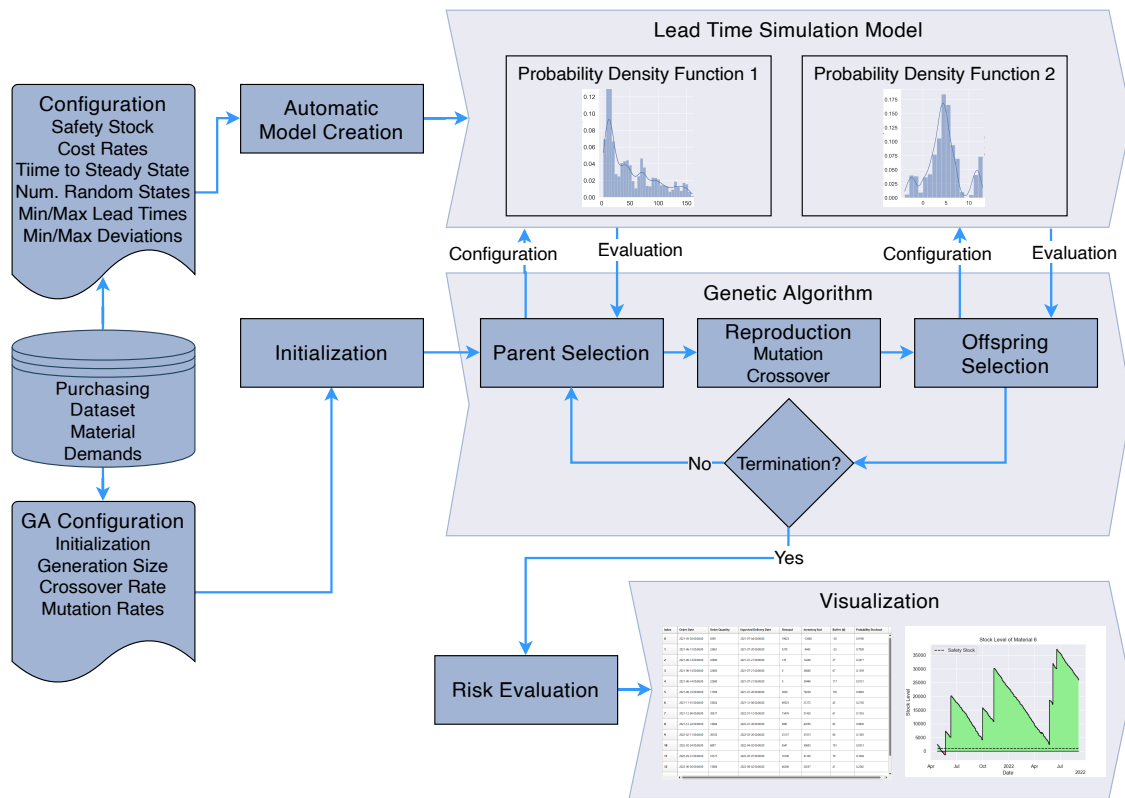


Figure 3. Conceptual purchasing algorithm based on a genetic algorithm and a simulation model.

model is automatically created whenever new historic data becomes available. The extremely variable lead times can result in different objective function values for each random state. Therefore, the user can define the number of observations for each solution candidate. The solution candidate then takes the mean inventory costs of all simulation runs as the fitness value.

5.2. Genetic Algorithm

The superordinate genetic algorithm uses the objective function of the purchasing problem as the fitness function and aims to minimize it. The representation has to represent solution candidates in a way that it can cover the entire solution space and that the operators are applicable without creating infeasible solutions. Regarding the purchasing problem, it is most suitable to include the decision variables in the representation as they have been defined in subsection 3.1. Each order consists of two genes, the reorder day and the order quantity. The reorder days can take integer values from within the planning horizon, whereas the order quantities can in theory take all positive integer values. The number of orders in the chromosome is unlimited. It is important that the two genes of one order

are never split. Hence, they form one coherent entity and behave like a single gene. The order of genes in the chromosome is irrelevant because each order has a defined time stamp respectively reorder day anyway.

The selection operator uses a tournament selection that runs multiple tournaments with subsets of the population. The tournament size determines the number of individuals in each tournament. The number of tournaments is equal to the generation size and each tournament selects the individual with the highest fitness which advances to the next generation.

The crossover operator uses a simple single-point crossover which determines a random uniformly distributed value and splits both parent chromosomes at that specific order index. Then, it recombines the chromosome pieces crosswise. Individuals are chosen for crossover with a certain crossover probability. Hence, it is possible that unmodified solution candidates advance to the next generation. When choosing two parent chromosomes with different lengths, it may happen that the crossover point is beyond the length of one of the chromosomes. The recombination works nevertheless, but the lengths of the offspring chromosomes will differ from the parents which adds additional flexibility to the genetic algorithm because it

is able to increase or decrease the number of orders.

After performing the crossover, the genes of the recombined solution candidates can mutate with a certain probability. The mutation operator has two stages, first it drops orders with a mutation drop probability and then adds new orders with a mutation add probability. Contrary to the crossover probability, the mutation probabilities apply to each gene and not to the entire chromosome. Hence, it is possible that the number of orders increases or decreases over time. Consequently, it is likely that the length of large solutions decreases, and the length of small solutions increases. However, it is recommended to set the same probability for both parameters to avoid drifting towards exceptionally large or small numbers of orders.

5.3. Risk Evaluation

After the optimization of the genetic algorithm terminates, it outputs the individual with the best fitness. Due to the multiple observations in different random states of each solution, the quality of the solution is validated to a certain extent, but the uncertainty is still high because the lead times remain variable and unpredictable. The risks of the recommended solution are not transparent to the decision-maker. Therefore, the risk evaluation module aims to counter the lack of transparency of metaheuristics.

The decision-makers of the company are especially concerned about possible stockouts in the case of late delivery. Therefore, a metric is required that determines the risk of stockouts in a transparent manner. The metric uses the lead time simulation model and the daily demands. The daily demands are known and fixed due to the scheduling algorithm that is always executed first in the optimization cycle. The lead time probability density function of each material is also known from the historic data and the same applies to the expected value of the probability density function. With these daily demands and the mean lead time, it is possible to compute the expected inventory development. Regarding the expected inventory development, there is still a certain inventory left in the warehouse for each order on the expected delivery day. If the order arrives late, the remaining inventory can still satisfy the daily demands until the warehouse is empty. We refer to this period as “buffer days”. The higher the buffer days, the more likely is it to compensate a late delivery without running out of stock. If the warehouse runs out of stock within an order cycle in the expected inventory development, the buffer days can also be negative. The risk evaluation module first calculates the buffer days for each order cycle.

To evaluate the risk of an order cycle, the metric additionally requires the cumulative distribution function of the kernel density estimation. When equating the cumulative distribution function with a specific value, it can determine the probability of a random value from the distribution being higher or lower than the initial value. This characteristic can be used to calculate the probability for delivery within the buffer days. The complementary probability is therefore the probability of running out of stock. Random values from the distribution always have a chance of 50% to be above or below the mean lead time. The buffer days shift the probability into one direction depending on their algebraic signs. If the buffer days are positive, the probability of stockout drops below 50% and vice versa if the buffer days are negative, the probability rises above 50%. In the end, the calculated probability indicates the risk for each order to cause an out-of-stock situation which is transparent for the decision-maker. With this information the decision-maker can identify critical orders and possibly adjust the recommended solution of the genetic algorithm.

6. Prototypical Implementation

The modules of the purchasing algorithm were implemented in the programming language Python using the latest version 3.9. As the architecture of the physical and information levels is not implemented yet, substitution modules are needed that create realistic input data for the purchasing algorithm. Therefore, we use a demand simulation model that generates a sequence of daily demands for the entire planning horizon. In addition, the prototypical implementation includes a “time to steady state” because the real supply network is in a steady state with several pending orders but the test environment has no information on pending orders. It rather starts in an idle state. The time to steady state describes the days at the beginning of the planning horizon which are excluded from the fitness calculation.

6.1. Purchasing Dataset

Creating the lead time and demand simulation models requires historic data. Therefore, the company provided a purchasing dataset which contains all the historic orders of one product from 2011 to 2020. The product is a printed circuit board for centrifuges in the medical sector that requires 17 different materials. Each material has exactly one supplier that always delivers from the same supplier site using the same transportation mode. The company orders rather high quantities but infrequently. Hence, there are only a few data points for each material. The dataset includes

in total 176 orders with order quantities, order dates, confirmed delivery dates and actual delivery dates. The lead times reach from 3 days to 315 days with the most data points between 10 and 70 days making the lead times extremely variable.

6.2. Results and Discussion

We only discuss the results of one exemplary material from the purchasing dataset. The planning horizon accounts for 500 days starting from the current day which is a typical duration for purchasing planning at the company. The time to steady state takes 50 days because the historic mean lead time of the associated supplier is 37 days. Therefore, it should be possible to reach a steady state within the period. The optimization run for the material on a standard PC requires 1:10 minutes and yields the recommended solution presented in Table 1. The first three order cycles have a high probability of stockout because the steady state is not reached yet, but however the first 50 days are not considered in the objective function. All the other order cycles have lower probabilities because of high inventory levels. With this transparent visualization of recommendations the decision-maker can quickly identify potential risks and adapt the suggested solution with his domain knowledge. Usually, the decision-maker has a much better information basis than the algorithm because the purchasing algorithm completely relies on historic data and statistical computation, whereas the decision-maker knows the circumstances. As the suppliers are often not able to supply materials, the decision-makers tend to order large amounts of material whenever it becomes available. Because of this behavior inventory levels are often higher than required. Decision-makers that are not accustomed to automatically created recommendations often distrust the outputs. Therefore, they still make decisions mainly based on their domain knowledge but can use the prototype as an additional resource for risk evaluation. After the remaining modules have been integrated in the production system, the tool will be ready for actual deployment and the decision-makers will be accustomed to the new process.

The tool also visualizes the inventory development chart for the recommended solution (see Figure 4). While the recommendations data frame considers the expected inventory development, the graph shows the inventory development of an arbitrary random state. Therefore, it can differ from the expected behaviour, but still it shows a possible future scenario. In this case, the graph does not include any stockouts, however the recommended solution anticipated low risks.

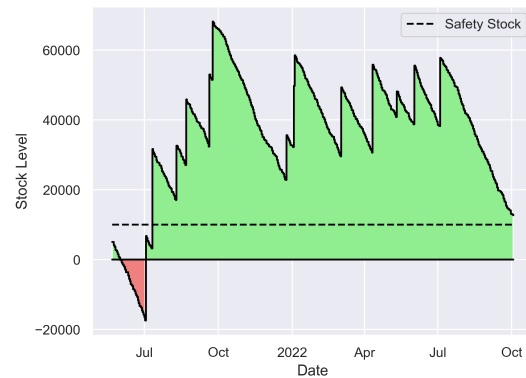


Figure 4. Expected inventory development of one exemplary material.

7. Conclusion and Future Research

The paper proposed a three-level concept for a DSS that automatically acquires data from a production system. In addition, we implemented a novel purchasing algorithm that can handle uncertain lead times and provides transparent and risk-aware recommendations. The lack of purchasing data is a significant issue in this supply chain planning problem which makes forecasting impossible. The only way to ensure permanent material availability in this environment is covering all possible outcomes from the probability density function of the lead time simulation model, and therefore maintaining high inventory levels. Improving the solution quality requires to reduce the lead time variability, for example by forecasting lead times or improving the collaboration with suppliers. The next step is fully implementing the conceptual DSS, and testing it in the real environment of the company. If the automatic data acquisition was implemented, the system could also generate more data for the algorithm development. Data-driven algorithms from the domain of machine learning seem to be especially suitable for the optimization cycle in terms of computation time.

References

- [1] A. Bennet and D. Bennet, "The decision-making process in a complex situation," in *Handbook on Decision Support Systems 1* (F. Burstein and C. W. Holsapple, eds.), pp. 3–20, Berlin, Heidelberg: Springer, 2008.
- [2] B. Fleischmann, H. Meyr, and M. Wagner, "Advanced planning," in *Supply Chain Management and Advanced Planning* (H. Stadler and C. Kilger, eds.), pp. 81–106, Berlin, Heidelberg: Springer, 2008.
- [3] S. Chopra and M. M. S. Sodhi, "Managing risk to avoid supply-chain breakdown," *MIT Sloan Management Review*, vol. 46, no. 1, pp. 53–61, 2004.
- [4] T. Ben Abid, O. Ayadi, F. Masmoudi, and A. M. Gil-Lafuente, "An integrated production-distribution

Table 1. Recommendations data frame with orders and risk evaluation.

Order Cycle	Order Date	Order Quantity	Expected Delivery Date	Demand	End of Cycle Inventory	Buffer Days	Stockout Probability
1	2021-05-23	29028	2021-06-29	20152	-15093	-28	0.8702
2	2021-06-16	24794	2021-07-23	13466	469	2	0.4793
3	2021-06-30	16317	2021-08-06	6424	18839	40	0.2307
4	2021-07-08	17564	2021-08-14	4139	31017	63	0.1479
5	2021-08-12	19389	2021-09-18	17073	31508	59	0.1640
6	2021-09-06	21559	2021-10-13	12194	38703	74	0.1160
7	2021-09-17	13569	2021-10-24	5824	54438	109	0.0357
8	2021-11-29	17705	2022-01-05	36892	31115	62	0.1516
9	2021-12-23	20636	2022-01-29	11988	36832	74	0.1160
10	2021-12-24	9231	2022-01-30	104	57364	112	0.0258
11	2022-03-03	25223	2022-04-09	35259	31336	61	0.1555
12	2022-04-11	20487	2022-05-18	20395	36164	72	0.1225
13	2022-04-30	8228	2022-06-06	9329	47322	91	0.0773

planning problem under demand and production capacity uncertainties: New formulation and case study,” *Mathematical Problems in Engineering*, vol. 2020, 2020.

- [5] B. Karimi, M. Ghare Hassanlu, and A. H. Niknamfar, “An integrated production-distribution planning with a routing problem and transportation cost discount in a supply chain,” *Assembly Automation*, vol. 39, no. 5, pp. 783–802, 2019.
- [6] S. M. Khalili, F. Jolai, and S. A. Torabi, “Integrated production–distribution planning in two-echelon systems: a resilience view,” *International Journal of Production Research*, vol. 55, no. 4, pp. 1040–1064, 2017.
- [7] G. Wang, “Integrated supply chain scheduling of procurement, production, and distribution under spillover effects,” *Computers & Operations Research*, vol. 126, p. 105105, 2021.
- [8] B. J. Park, H. R. Choi, and M. H. Kang, “Integration of production and distribution planning using a genetic algorithm in supply chain management,” in *Analysis and Design of Intelligent Systems using Soft Computing Techniques* (P. Melin, O. Castillo, E. G. Ramírez, J. Kacprzyk, and W. Pedrycz, eds.), pp. 416–426, Berlin, Heidelberg: Springer, 2007.
- [9] S. A. Torabil and E. Hassini, “Multi-site production planning integrating procurement and distribution plans in multi-echelon supply chains: an interactive fuzzy goal programming approach,” *International Journal of Production Research*, vol. 47, no. 19, pp. 5475–5499, 2009.
- [10] A. P. Kanyalkar and G. K. Adil, “A robust optimisation model for aggregate and detailed planning of a multi-site procurement-production-distribution system,” *International Journal of Production Research*, vol. 48, no. 3, pp. 635–656, 2010.
- [11] R. V. Ramasesh, “Procurement under uncertain supply lead times — a dual- sourcing technique could save costs,” *Engineering Costs and Production Economics*, vol. 21, no. 1, pp. 59–68, 1991.
- [12] O. B. Ammar, A. Dolgui, F. Hnaien, and M. A. Louly, “Supply planning and inventory control under lead time uncertainty: A review,” *IFAC Proceedings Volumes*, vol. 46, no. 9, pp. 359–370, 2013.
- [13] N. Salehi Sadghiani, S. A. Torabi, and N. Sahebjamnia, “Retail supply chain network design under operational and disruption risks,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 75, pp. 95–114, 2015.
- [14] S. Fallah-Jamshidi, N. Karimi, and M. Zandieh, “A hybrid multi-objective genetic algorithm for planning order release date in two-level assembly system with random lead times,” *Expert Systems with Applications*, vol. 38, no. 11, pp. 13549–13554, 2011.
- [15] F. Hnaien, X. Delorme, and A. Dolgui, “Genetic algorithm for supply planning in two-level assembly systems with random lead times,” *Engineering Applications of Artificial Intelligence*, vol. 22, no. 6, pp. 906–915, 2009.
- [16] R. Wittrock, “Scheduling parallel machines with major and minor setup times,” *International Journal of Flexible Manufacturing Systems*, vol. 2, no. 4, pp. 329–341, 1990.
- [17] W. Kritzing, M. Karner, G. Traar, J. Henjes, and W. Sihn, “Digital twin in manufacturing: A categorical literature review and classification,” *IFAC PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.
- [18] S. Lang, T. Reggeline, J. Schmidt, M. Müller, and A. Nahhas, “Neuroevolution of augmenting topologies for solving a two-stage hybrid flow shop scheduling problem: A comparison of different solution strategies,” *Expert Systems with Applications*, vol. 172, no. 114666, 2021.
- [19] A. Nahhas, P. Aurich, T. Reggeline, and J. Tolujew, “Heuristic and metaheuristic simulation-based optimization for solving a hybrid flow shop scheduling problem,” in *Proceedings of the 2016 International Conference on Modeling and Applied Simulation* (A. Bruzzone, F. De Felice, C. Frydman, M. Massei, Y. Merkuriev, and A. Solis, eds.), 2016.
- [20] B. Rolf, T. Reggeline, A. Nahhas, S. Lang, and M. Müller, “Assigning dispatching rules using a genetic algorithm to solve a hybrid flow shop scheduling problem,” *Procedia Manufacturing*, vol. 42, pp. 442–449, 2020. International Conference on Industry 4.0 and Smart Manufacturing (ISM 2019).